Министерство науки и ВЫСШЕГО образования   
Российской Федерации

Федеральное автономное образовательное учреждение высшего образования  
«Санкт-Петербургский государственный политехнический   
университет Петра Великого»  
**Институт среднего профессионального образования**

**Лабораторная работа № 10  
Тема:** Создание проектов с визуальными элементами пользовательского интерфейса: списками, редакторами.

**Вариант 9**

Выполнил: студент группы 32919/8  
Крупина Елена

Проверил: преподаватель   
Прокофьев А.А.

Санкт-Петербург  
2024

**Цель работы:** получить практические навыки создания элементов пользовательского интерфейса со списками, редакторами, и разработки методов, реализующих работу с такими элементами интерфейса.

**Задание:**

1. Прочитайте свой вариант задания (см. ниже). Запишите в отчёт, что дано и что надо найти, с указанием типов исходных данных и результатов.
2. . Разработайте описание класса в соответствие с условием задачи вашего варианта. Для добавления нового класса в проект выберите пункт меню «Проект» → «Добавить класс». Новый класс будет добавлен в отдельном программном модуле. Включите в описание класса следующие методы: конструктор, перегрузку метода ToString() для формирования строки для вывода данных экземпляра класса, свойство для чтения значения скрытого поля, необходимого для расчёта.
3. Продумайте, какие элементы интерфейса необходимо разместить на форме, чтобы обеспечить удобную и наглядную работу пользователя. Интерфейс должен обязательно включать список listBox или comboBox и многострочный редактор richTextBox. Для ввода некоторых данных можно использовать поле ввода по маске или переключатель значений numbericUpDown, для ввода даты – компонент

календарь.

1. Настройте элементы интерфейса. Скопируйте в отчёт скриншот формы на этапе конструирования и надпишите компоненты с указанием их типов.
2. План испытаний

**Ход работы:**

1. Прочитайте свой вариант задания (см. ниже). Запишите в отчёт, что дано и что надо найти, с указанием типов исходных данных и результатов.

**Вариант 9**

Разработать программу ввода названий товаров, их цены и количества на складе,

добавления, удаления, изменения этих сведений в списке, сохранения списка в

текстовый файл, вычисления средней стоимости перечисленных товаров и поиска

тех товаров, которых осталось меньше всего на складе.

Таблица 1 – Условие

|  |  |  |
| --- | --- | --- |
| Дано: | Наименование товара | string |
| Цена | double |
| Количество | int |
| Найти: | Средняя стоимость всех товаров | double |
| Товар, которого меньше всего на складе | Экземпляр класса Товары |

1. . Разработайте описание класса в соответствие с условием задачи вашего варианта. Для добавления нового класса в проект выберите пункт меню «Проект» → «Добавить класс». Новый класс будет добавлен в отдельном программном модуле. Включите в описание класса следующие методы: конструктор, перегрузку метода ToString() для формирования строки для вывода данных экземпляра класса, свойство для чтения значения скрытого поля, необходимого для расчёта.

public class items

{

private string name;

private double price;

private int count;

public items()

{

this.name = string.Empty;

this.price = 0;

this.count = 0;

}

public string Name

{

get { return name; }

set { name = value; }

}

public double Price

{

get { return price; }

set { price = value; }

}

public int Count

{

get { return count; }

set { count = value; }

}

public override string ToString() {

return $"{name} {price}₽ {count}\n";

}

}

1. Продумайте, какие элементы интерфейса необходимо разместить на форме, чтобы обеспечить удобную и наглядную работу пользователя. Интерфейс должен обязательно включать список listBox или comboBox и многострочный редактор richTextBox. Для ввода некоторых данных можно использовать поле ввода по маске или переключатель значений numbericUpDown, для ввода даты – компонент

календарь.

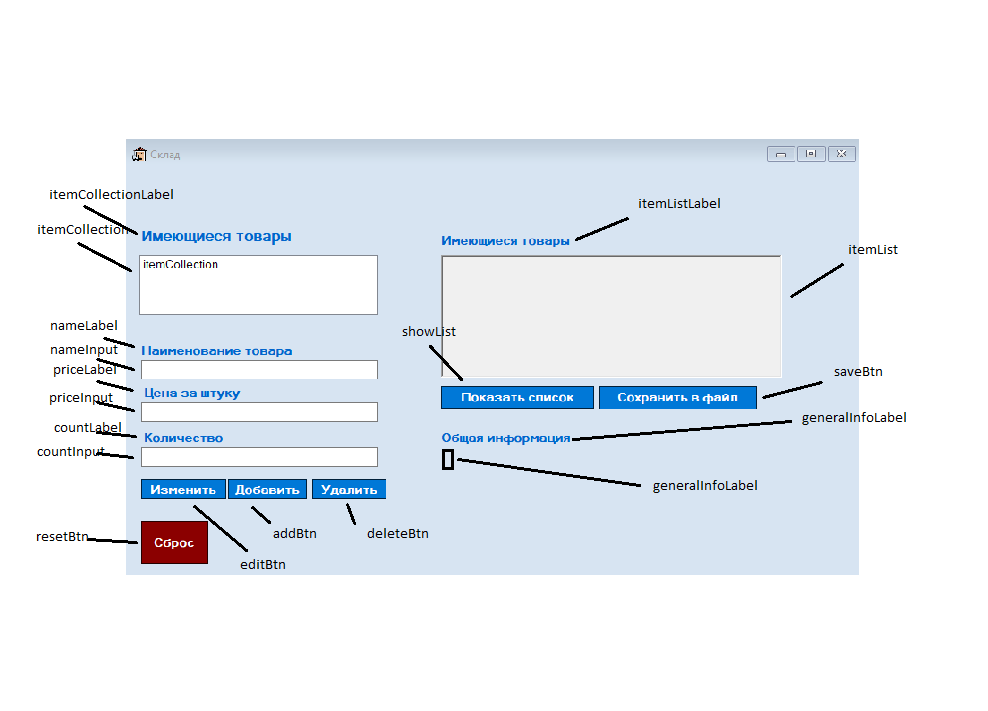


Рис.1 – Вид пользовательского интерфейса

1. Настройте элементы интерфейса. Скопируйте в отчёт скриншот формы на этапе конструирования и надпишите компоненты с указанием их типов.

Таблица 2 – Компоненты с указанием типов

|  |  |
| --- | --- |
| Имя | Тип |
| itemListLabel | Label  Надпись |
| itemList | RichTextBox  Текстовый редактор |
| showList | Button Кнопка |
| saveBtn | Button Кнопка |
| generalInfoLabel | Label  Надпись |
| itemCollectionLabel | Label  Надпись |
| itemCollection | ListBox Список |
| nameLabel | Label  Надпись |
| nameInput | Label  Надпись |
| priceLabel | Label  Надпись |
| priceInput | TextBox Поле ввода |
| priceLabel | Label  Надпись |
| priceInput | TextBox Поле ввода |
| countLabel | Label  Надпись |
| countInput | TextBox Поле ввода |
| editBtn | Button Кнопка |
| addBtn | Button Кнопка |
| deleteBtn | Button Кнопка |
| resetBtn | Button Кнопка |

**Реализация функцинала интерфейса:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace LAB10

{

public partial class Storehouse : Form

{

public Storehouse()

{

InitializeComponent();

}

//Ограничение полей ввода---------------------------------------------

private void priceInput\_KeyPress(object sender, KeyPressEventArgs e)

{

// Проверяем, является ли нажатая клавиша цифрой или точкой

if ((e.KeyChar >= '0' && e.KeyChar <= '9') || e.KeyChar == '.'|| e.KeyChar== (char)Keys.Back)

{

// Проверяем, есть ли уже точка в тексте

if (e.KeyChar == '.' && (priceInput.Text.Length == 0 || priceInput.Text.Contains('.')))

{

e.Handled = true; // Игнорируем ввод, если точка первая или уже есть

}

}

else

{

e.Handled = true; // Игнорируем другие символы

}

}

private void countInput\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar >= '0') && (e.KeyChar <= '9' ) || e.KeyChar == (char)Keys.Back)

{ return; }

else { e.Handled = true; }

}

private void nameInput\_KeyPress(object sender, KeyPressEventArgs e)

{

if (char.IsLetter(e.KeyChar) || e.KeyChar == (char)Keys.Back)

{

return;

}

else { e.Handled = true; }

}

//Двойное нажатие на элемент списка--------------------------------------

private void itemCollection\_DoubleClick(object sender, EventArgs e)

{

string[] item = itemCollection.SelectedItem.ToString().Split(' ');

nameInput.Text = item[0];

priceInput.Text = item[1];

countInput.Text = item[2];

}

List<items> list = new List<items>();

//Добавление элемента в список------------------------------------------

private void addBtn\_Click(object sender, EventArgs e)

{

items item = new items();

item.Name = nameInput.Text;

try

{

if (double.TryParse(priceInput.Text, out double price))

{

item.Price = Convert.ToDouble(priceInput.Text);

if (int.TryParse(priceInput.Text, out int count))

{

item.Count = count;

itemCollection.Items.Add(item);

list.Add(item);

}

else { throw new FormatException(); }

}

else { throw new FormatException(); }

}

catch {MessageBox.Show("Некоррректный ввод. Попробуйте снова."); }

}

//Удаление элемента из списка------------------------------------------

private void deleteBtn\_Click(object sender, EventArgs e)

{

itemCollection.Items.RemoveAt(itemCollection.SelectedIndex);

list.RemoveAt(itemCollection.SelectedIndex);

}

//Изменение элемента списка---------------------------------------------

private void editBtn\_Click(object sender, EventArgs e)

{

// Проверяем, что выбранный элемент не null и является экземпляром класса items

if (itemCollection.SelectedItem is items selectedItem)

{

// Обновляем поля выбранного элемента

selectedItem.Name = nameInput.Text;

// Пытаемся преобразовать вводимые данные в цену и количество

if (double.TryParse(priceInput.Text, out double price))

{

selectedItem.Price = price;

}

else

{

MessageBox.Show("Некорректный ввод цены. Попробуйте снова.");

return;

}

if (int.TryParse(countInput.Text, out int count))

{

selectedItem.Count = count;

}

else

{

MessageBox.Show("Некорректный ввод количества. Попробуйте снова.");

return;

}

// Обновляем отображение в ListBox

itemCollection.Items[itemCollection.SelectedIndex] = selectedItem;

}

else

{

MessageBox.Show("Выберите элемент для редактирования.");

}

}

//Перевод содержимого ListBox в RichTextBox

private void showList\_Click(object sender, EventArgs e)

{

itemList.Clear();

for (int i = 0; i < itemCollection.Items.Count; i++)

{

itemList.AppendText( itemCollection.Items[i].ToString());

}

double totalPrice = 0;

for (int j = 0; j < list.Count; j++)

{

totalPrice += list[j].Price;

}

double averagePrice = totalPrice / list.Count;

double minCount = 0;

items minItem=new items();

minItem.Count = int.MaxValue;

for (int k = 0; k < list.Count; k++)

{

if (list[k].Count < minItem.Count)

{

minItem=list[k];

}

}

generalInfoOutput.Text = $"Средняя стоимость перечисленных товаров: {averagePrice}₽\n Товар,которого меньше всего на складе: {minItem}";

}

//Кнопка сброса всего вообще короче--------------------------------------------

private void resetBtn\_Click(object sender, EventArgs e)

{

itemList.Clear();

itemCollection.Items.Clear();

nameInput.Text=string.Empty;

priceInput.Text=string.Empty;

countInput.Text=string.Empty;

list.Clear();

}

//Сохранение файла-----------------------------------------------------------

private void saveBtn\_Click(object sender, EventArgs e)

{

// Создаем диалоговое окно для сохранения файла

using (SaveFileDialog saveFileDialog = new SaveFileDialog())

{

saveFileDialog.Filter = "Rich Text Format (\*.rtf)|\*.rtf|Text Files (\*.txt)|\*.txt|All Files (\*.\*)|\*.\*";

saveFileDialog.Title = "Сохранить текст в файл";

// Если пользователь выбрал имя файла и нажал OK

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

try

{

// Сохраняем содержимое RichTextBox в выбранный файл

itemList.SaveFile(saveFileDialog.FileName, RichTextBoxStreamType.RichText);

MessageBox.Show("Файл успешно сохранен!", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при сохранении файла: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

}

}

Таблица 3 – План испытаний

|  |  |  |  |
| --- | --- | --- | --- |
| Проверяемые  требования | Сообщения программы и вводимые данные | Ожидаемые результаты | Фактически  е результаты |
| Способность  программы  обеспечить ввод  исходных данных | Ввести наименование товара без пробелов, специальных знаков и цифр (тротил)  Ввести цену используя только цифры без пробелов и спец. символов, разделитель – точка (12341)  Ввести количество целое положительно число без пробелов и спец. символов.  (1) | Вводимые значения должны отображаться в полях ввода | Пройден см. Рис.2 |
| Способность  программы  обеспечить  контроль вводимых  данных | Попробовать ввести в наименовании числа и спец. символы. (трт1123 ?:) В поле цена вместо точки использовать запятую (123,121), попробовать ввести минус (-1) | В первом поле допустимы только буквы , клавиша BackSpace.  Во втором поле допустимы цифры и одна точка.  В третьем поле допустимы только цифры  Недопустимые символы клавиатуры блокируются и не отображаются в полях ввода | Пройден см. Рис.3 |
| Способность  программы  обеспечить  контроль пустых  полей ввода | Попробовать оставить  одно поле ввода пустым и  нажать на кнопку  «Добавить». | Сообщение об ошибке “Поля не должны быть пустыми”. После закрытия окна можно продолжать работу с приложением | Пройден см. Рис.4 |
| Способность  программы  добавлять данные в  список и  производить расчет  среднего возраста  сотрудников и  обеспечивать  вывод результатов  на экран | Ввести наименование (строка и без пробелов и спец. символов), цену(положительное число с разделителем точкой) и количество (целое положительное число)  Нажать кнопку Добавить затем нажать кнопку Показать список | Элемент списка появляется в поле списка Имеющиеся товары, строка появляется в поле многострочного редактора.  Отображается общая информация, средняя цена посчитана правильно, товар с наименьшим количеством на складе выбран правильно. | Пройден см. Рис.5 |
| Способность  программы  производить сброс  списка | Нажать на кнопку “Сброс” продолжить вводить данные о новых товарах. | Очищаются все поля ввода, список и многострочный редактор. Общая информация обнуляется. | Пройден см. Рис.6 |
| Способность  программы  выполнять расчёт  среднего возраста  после сброса | После нажатия на кнопку  «Сброс» продолжить  вводить данные о новых  сотрудниках. | Наименование, количество и цена добавляются в список и многострочный редактор, каждый товар с новой строки. Общая информация считается заново. | Пройден см. Рис.7 |

**Результаты испытаний:**

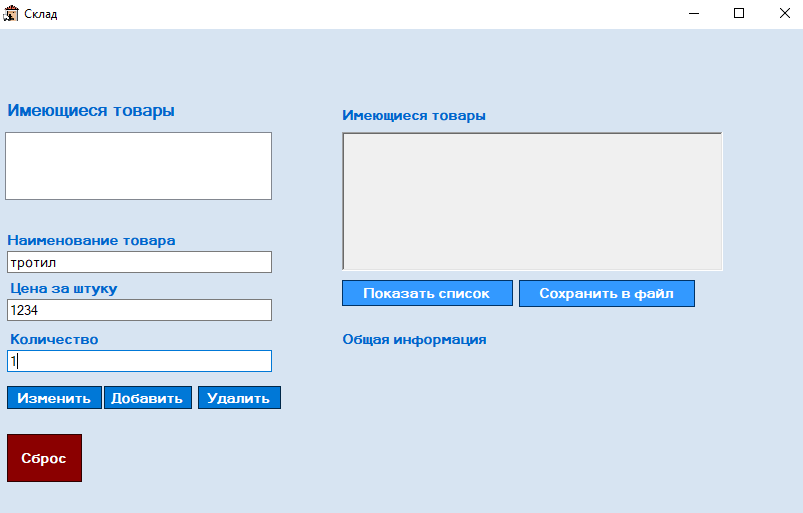


Рис.2 – Результат испытания

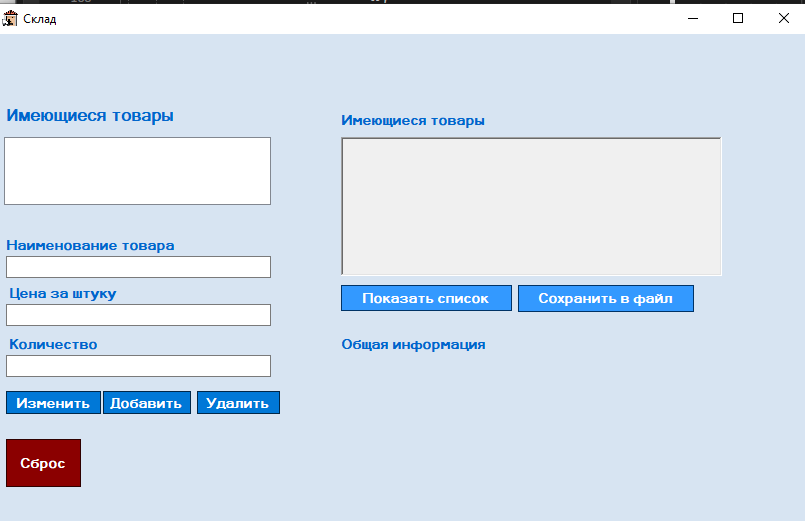


Рис.3 – Результат испытания

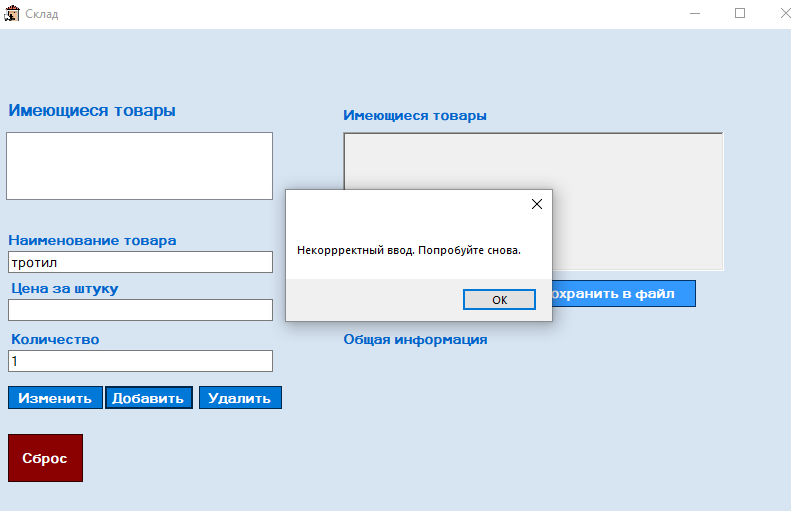


Рис.4 – Результат испытания

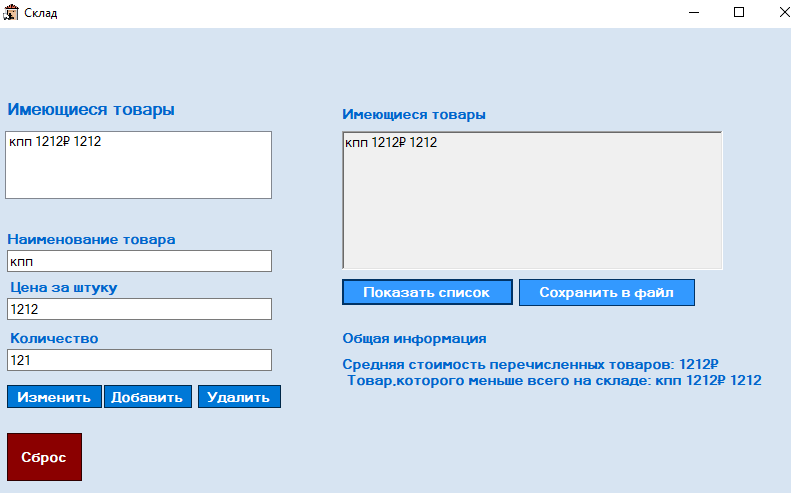


Рис.5 – Результат испытания

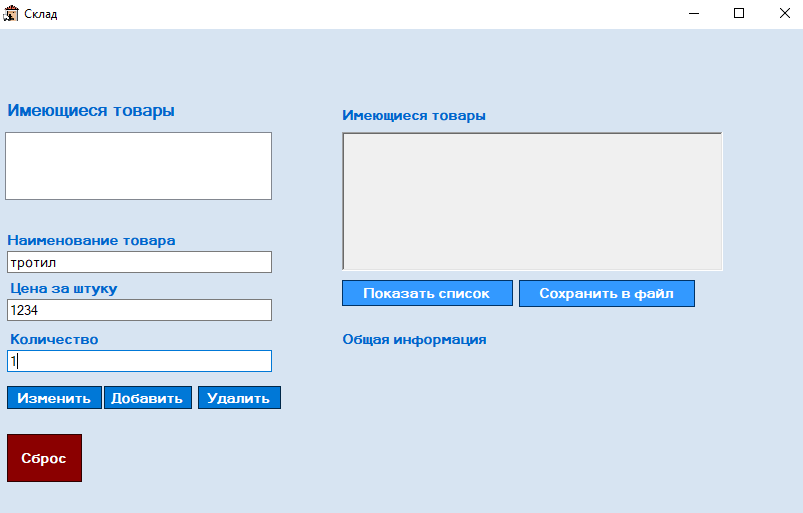


Рис.6 – Результат испытания

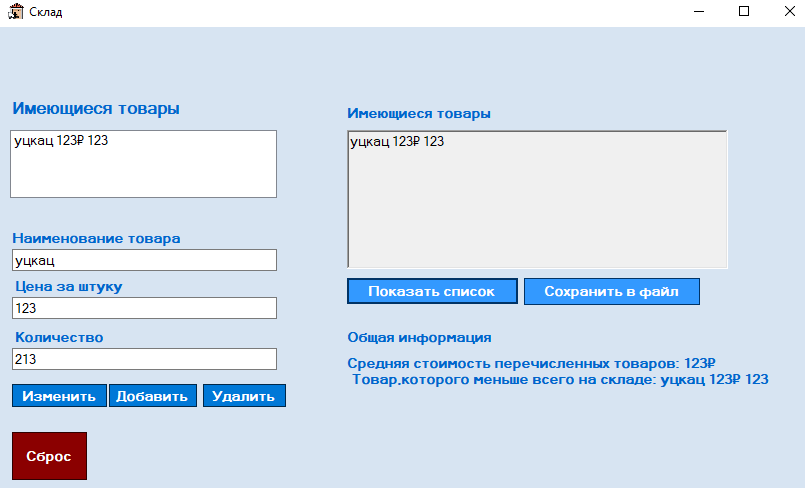


Рис.7 – Результат испытания